A Language Modeling Approach for the Classification of Audio Music

Gonçalo Marques and Thibault Langlois

DI–FCUL TR–09–02

February, 2009

HCIM - LaSIGE
Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa
Campo Grande, 1749–016 Lisboa
Portugal

Technical reports are available at `http://www.di.fc.ul.pt/tech-reports`. The files are stored in PDF, with the report number as filename. Alternatively, reports are available by post from the above address.

Abstract. The purpose of this paper is to present a method for the classification of musical pieces based on a language modeling approach. The method does not require any metadata and is used with raw audio format. It consists in 1) transforming music data into a sequence of symbols 2) building a model for each category by estimating n-grams from the sequences of symbols derived from the training set. The results obtained on three audio datasets show that, providing the amount of data is sufficient for estimating the transitions probabilities of the model, the approach performs very well. The performance achieved with the ISMIR 2004 Genre classification dataset is, to our knowledge, one of the best published in the literature.

1 Introduction

The task of automatic genre classification, based solely on the audio contents of music signals, is a challenging one. Genre classification is not by any means consensual, even when performed by human experts. This is partly due to the complexity of music signals: a given song can be a mix of several genres. Therefore, it is not possible to achieve 100% accuracy in a classification system. Additionally, audio signals are not suited to be directly fed into a classification system, therefore some alternate, more compact representation is needed. Typically, some audio characteristics are extracted, such as timbre, chroma, chords, rhythm, melody or chorus. Nevertheless, it is difficult to combine the resulting features, since they often have different time scales.

Despite the complexity of the problem, techniques for music genre classification have attracted considerable attention in recent years (see for instance [1, 2] and references therein).

The most common approach to genre classification of audio music signals is to divide the signal in short overlapping frames (generally $10 - 100$ms with a 50% overlap), and some features, usually based on the spectral representation of the frame are extracted (*eg* MFCCs, spectral spread, rolloff, centroid, etc). After this process, each music signal is represented by a sequence of feature vectors that can be thought of as samples from a distribution, which can be modeled by various techniques. Similarly, the distributions of the classes can be estimated by grouping songs of the same genre. For instance, k-means [3], or gaussian mixture models (GMM) [2, 4, 5] can be used to model the class distributions. Once the models are obtained, one can use the "bag of frames" classifiers [6], compare models using a Earth Mover's distance [3], or use the Kullback-Leibler divergence [2]. The main drawback with this type of approaches is that only the short-time characteristics of the signal are modeled. It does not take into account the ordering of the feature vectors, and therefore, the dynamics are discarded. To overcome this limitation several authors complement the short-time features with other sets of features that model the dynamics of the audio signal. Rhythmic features are a typical example [5, 7], but other long-term descriptors such as the fluctuation patterns in [4] can be used.

Another approach is to aggregate several short-time frames in larger scale windows (usually a few seconds) in order to capture the long-term dynamics. For example [5, 6, 8, 7] model temporal variations by calculating some statistics of the short time features over longer temporal windows. Some authors report a significant improvements in classification accuracy when the long-term windows are used, although the work of Aucouturier and Pachet [9] contradicts this result.

In this work, we use a language model approach to classify music signals in different genres. Our method is similar to Chen et al. [10] in some aspects. They propose to use a text categorization technique to perform musical genre classification. They build a HMM from the MFCC coefficients using the whole database. The set of symbols is represented by the states of the HMM. Music symbols are tokenized by computing 1 and 2-grams. The set of tokens is reduced using Latent Semantic Indexing.

The outline of this paper is as follows. In section 2 we describe the language model approach. In section 3 the feature extraction and classification processes for audio files are explained. In the following two sections the results obtained with audio signal databases are evaluated. We close with some final conclusions and future work.

2 A language modeling approach

The idea behind our proposal is to use language modeling techniques[11] for the classification of music in audio format. In order to use this kind of approach we have to:

1. build a dictionary of symbols that are used to represent any song;
2. define a procedure which transforms a song into a sequence of symbols;
3. build a model for each category of music;
4. Find a procedure which, from a set of models and a sequence, determine the best model that fits this sequence.

3 Classification of Audio files

3.1 Two-stage clustering

For audio files we use classical twelve Mel Frequency Cepstrum Coefficients (MFCC) as the only feature[1]. The first step consists in extracting the most representative frames for each song of the training set. This is done using the k-means clustering algorithm. The same value for the k parameter is used for every piece of music of the training set. We call k_1 the number of clusters per music used in this phase. We obtain $n \times k_1$ vectors where n is the number of songs of the training set. Let us call $\mathcal{F}_1$ this set.

The second step consists in finding a set of representative frames in $\mathcal{F}_1$. Again, we use the k-means clustering algorithm. Let's call $\mathcal{F}_2$ the set of k_2

[1] All audio files were sampled at 22050 Hz, mono and frame duration of 93ms.

centroids obtained from the clustering. A symbol is assigned to each centroid. The dictionary $\mathcal{D}$ is therefore composed of k_2 symbols.

The procedure used to transform a song into a sequence of symbols is as follows: 1. Compute the MFCC. 2. For each frame compute the 1st nearest neighbor in $\mathcal{F}_2$ and assign the corresponding symbol of the dictionary.

Thanks to this two stage approach, our algorithm is very scalable. We can process the whole music database and use the sets of k_1 centroids as a compact representation of musics. Several music genre models can be build based on this representation. This aspect contrasts from the approach proposed by Chen [10] where the whole set of MFCC frames are used to build a HMM for each genre.

3.2 Estimation of n-grams

The following phase is the estimation of a language model for each category into which we want to classify the songs.

For each music category, the probability of each bi-gram is computed by processing every sequence of symbols and counting the occurrences of the symbols transitions. The result is a transition probability matrix that contains, for each pair of symbols (s_i, s_j), the probability $P(s_j|s_i)$ of symbol s_i to be followed by the symbol s_j. In the context of a genre classification task, a model, represented by a transition probability matrix is estimated for each genre by processing the n-grams of the files that belong to each genre.

After this estimation, the probability of many transitions is zero which is not desirable. Indeed the training sets used to estimate the models are finite and small. Without modification, if a single transition that has not been seen before in the training set is observed in a test sequence, the probability that the sequence belongs to the model would automatically be zero. In order to avoid this zero-frequency problem, the model is smoothed by adding a small constant $\epsilon = 1.0e - 5$ to each transition that has not been observed in the data set.

3.3 Classification of music files

The classification of a music file is done by transforming the music into a sequence of symbols and computing the probability that each model would generate this sequence. Given a model M, the probability that it would generate the sequence $S = s_1, s_2, ...s_n$ is:

$$P_M(s_{i=1..n}) = P_M(s_1) \prod_{i=2}^{n} P_M(s_i|s_{i-1}) \tag{1}$$

which is better calculated as

$$P_M(s_{i=1..n}) = \log(P_M(s_1)) + \sum_{i=2}^{n} \log(P_M(s_i|s_{i-1})) \tag{2}$$

This score is computed for each model M_j and the class corresponding to the model that maximize the score values is assigned to the sequence of symbols.

The approach described in this paper can be seen as a set of Vector Quantization-based Markov Models built for each category to be classified. The following sections describe some results obtained with this technique on various datasets.

4 Results

4.1 ISMIR 2004 Genre Classification

We used two different datasets to evaluate our method. The first one is the ISMIR 2004 genre classification dataset which is composed of six musical genres with a total of 729 songs for training and 729 songs for test[2].

k_1	k_2	% correct
10	100	79.29
20	150	79.15
30	150	80.52
20	200	80.11
30	200	**80.52**
40	200	80.11
20	300	79.84
30	300	79.97
20	400	80.25
40	400	79.42

Table 1. Percentage of correctly classified songs on the test set, for various k_1 and k_2 parameter values.

CLASSICAL	300	1	0	0	0	19	93.75%
ELECTRONIC	2	95	1	1	7	8	83.33%
JAZZBLUES	0	3	14	0	6	3	53.85%
METALPUNK	0	0	0	20	23	2	44.44%
ROCKPOP	2	15	0	4	76	5	74.51%
WORLD	12	16	0	0	12	82	67.21%

Table 2. Confusion matrix obtained with the best result of Table 1. The last column correspond to the percentage of correctly classified song for each genre.

Table 1 shows the percentage of correctly classified songs in the test set for various k_1 and k_2 parameter values. The best result (80.52%) is detailed in

[2] The distribution of songs along the six genres is: classical: 319; electronic: 115 jazzblues: 26; metalpunk: 45; rockpop: 101; world: 123 for the training and the test set.

Table 2 where the confusion matrix is shown. This result must be compared to the results obtained by the participants of the ISMIR 2004 Genre classification Challenge[3] and the results published thereafter. Pampalk et al. [4] obtained 84.07% and Annesi et al. [12] obtained 82.10%. If we weight the percentages with the prior probability of each class Pampalk obtains a 78.78% and we obtain 80.53%. Even if we do not obtain the best results for every evaluation metric, the results are interesting especially if we take into account that only simple spectral-based features are used. However, as noted by Aucouturier, we may be reaching a "glass ceiling" in this case.

4.2 Our dataset

The second dataset was made by us. It is composed of 7 genres: Jazz, Rock'n'Roll, Bossanova, Punk, Fado, Oriental, and Classical. We chose artists/albums that belong to each genre without ambiguity:

Jazz: Dave Brubeck, Duke Ellington, John Coltrane, Miles Davis, Thelonious Monk and Louis Armstrong (110 songs).

Rock'nRoll: Bill Haley, Chuck Berry, Jerry Lee Lewis, Little Richard and The Shadows (167 songs).

Bossa Nova: António Carlos Jobim, Dori Caymmi and João Gilberto (110 songs).

Punk: Bad Religion, Buzzcocks, Down by Law, No Fun at All and Sham 69 (158 songs).

Fado: Ana Moura, Camané, Carlos do Carmo, Mafalda Arnauth and Mariza (109 songs).

Oriental: Anouar Brahem, Rabih Abou-Khalil and Ravi Shankar (88 songs featuring traditional oriental string instruments such as esraj, sarangi and percussions).

Classical: Several Piano Concertos, from the Romantic Period, by: Moscheles, Pierné, Parry, Stanford, Mendelssohn, Vianna da Motta, Balakirev, Rimsky-Korsakov, Alkan, Henselt and Kalkbrenner (69 songs).

Although the albums were chosen for being homogeneous in their musical style, there are exceptions, for example blues songs in a Rock'n'Roll album. These exceptions were **not** removed from the dataset. In the first set of experiments, we split every album in two, keeping the first half of the songs for the training set and the second half for test. The Table 3 shows the percentage of correctly classified songs on the test set. One can see the (little) sensibility of the algorithm with respect to a wide range of the parameters k_1 and k_2. A typical confusion matrix is shown in Table 4.

We made a second set of experiments where 50% of the whole dataset was randomly selected for training. When repeating ten times this experiment (using $k_1 = 20$ and $k_2 = 200$) we obtain a mean success rate of 87.52% with a standard deviation equal to 1.87.

One of the reasons to constitute our own dataset was to be able to study the influence of various aspects. One of this aspects is whether the classifier is doing artist identification instead of genre classification. Pampalk [4] recommends using

[3] See http://ismir2004.ismir.net/genre_contest/results.htm.

k_1	k_2	% correct
10	25	74.88
10	50	81.03
10	100	86.21
20	100	84.98
20	200	85.71
20	300	86.70
20	400	86.95

Table 3. Percentage of correctly classified songs for various k_1 and k_2 values.

JAZZ	44	6	2	0	0	4	0	78.57%
ROCKNROLL	2	76	1	4	0	1	0	90.48%
BOSSANOVA	3	2	47	0	2	1	0	85.45%
PUNK	1	20	0	58	0	0	0	73.42%
FADO	0	0	0	0	54	0	0	100.00%
ORIENTAL	4	3	0	0	0	37	0	84.09%
CLASSICAL	2	0	0	0	0	0	32	94.12%

Table 4. The confusion matrix obtained when using $k_1 = 20$ and $k_2 = 200$. The last column shows the success rate for each class.

Artist Filtering[4] (AF) in order to avoid this problem. We did a set of experiments with AF by selecting an artist for the test set of each genre while keeping the other artists for the training set. Repeating eight times we get an average success rate of 65% with a standard deviation of 4.37. These results confirms those described in [4]. The success rate is significantly lower *on average* than without AF, but if we look at the best performance, 71% of the songs are correctly classified. While doing these experiments, we learned a few lessons:

Our approach consists in building a model based on a representation of timbres (and probability transitions between these timbres). The approach is not immune to over-fitting but we think that failures are due mainly to the absence, in the training set, of a kind of timbre that is relevant to the musical genre we want to model. This is only partially related to the artist.

Certain conditions adversely affect our method. For example, when leaving the Bossa Nova artist João Gilberto in the test set, one of his albums was completely misclassified. It was a live recording with significant sequences of applause and speech. We believe this is the reason why it was not correctly classified. To be correctly classified we would need other live recordings in the training set.

Our method needs a large amount of data because it needs to collect representative samples of timbre that characterize a genre and estimate the probability transitions as closely as possible.

[4] Artist Filtering consist building the datasets such that no artists appear in both training and test sets.

For illustrations purpose we show a picture of the transition probability matrix obtained for the Jazz-Blues genre of the ISMIR 2004 dataset on Figure 1. Each pixel represents $\log(P(s_j|s_i))$ (the quantity that is summed in equation 2). The diagonal with white pixels represents the transition probability to the same symbol (which is high), gray horizontal lines correspond to symbols that are very rarely found in that style and the other gray pixels show the contribution of the corresponding transitions towards the identification of the genre.

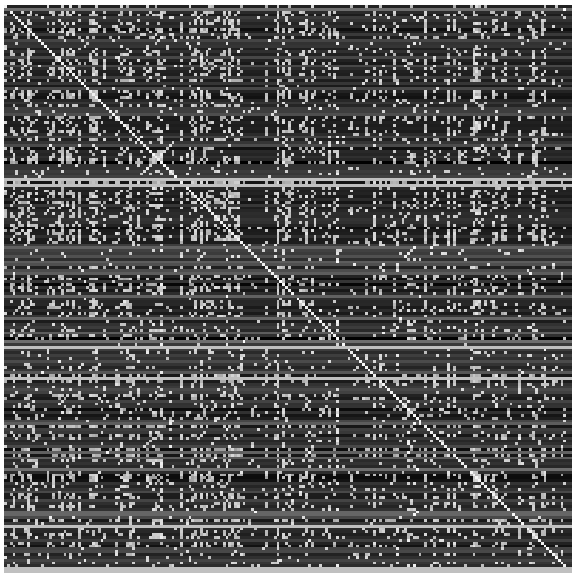

Fig. 1. The transitions probabilities matrix obtained for the JAZZBLUES genre of the ISMIR 2004 dataset.

5 Conclusion and Future Work

We proposed a genre classification framework for music files, based on a language modeling approach. Experiments on audio music signals show the potential of the method. Our system performs well, especially if we take into account the simplicity of the features used. Also, it is worth noting that the classifier accuracy is not significantly affected by the values of k_1 and k_2. In this work, due the size limitations of the datasets, we only estimated the probability of bi-grams, but we intend to build larger datasets to be able to estimate the transitions probabilities of three or more consecutive elements of the feature sequences. In the future, we also intend to experiment the use of vector quantization-based and continuous-density HMMs to model music genres.

References

1. Berenzweig, A., Logan, B., Ellis, D., Whitman, B.: A large-scale evaluation of acoustic and subjective music similarity measures. Computer Music Journal **28**(2) (2004) 63–76
2. Aucouturier, J.J., Pachet, F.: Improving timbre similarity: How high is the sky? Journal of Negative Results in Speech and Audio Sciences **1**(1) (2004)
3. Logan, B., Salomon, A.: A music similarity function based on signal analysis. In: ICME. (2001)
4. Pampalk, E., Flexer, A., Widmer, G.: Improvements of audio-based music similarity and genre classification. In: ISMIR. (2005)
5. Tzanetakis, G., Cook, P.: Musical genre classification of audio singals. IEEE Trans. on Speech and Audio Processing **10**(5) (2002) 293–302
6. West, K., Cox, S.: Finding an optimal segmentation for audio genre classification. In: ISMIR. (September 2005)
7. Lidy, T., Rauber, A.: Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In: ISMIR. (2005) 34–41
8. Bergstra, J., Casagrande, N., Erhan, D., Eck, D., Kégl, B.: Aggregate features and AdaBoost for music classification. Machine Learning **65**(2-3) (2006) 473–484
9. Aucouturier, J.J., Pachet, F.: Improving timbre similarity: How high is the sky? Pattern Recognition Letters **28**(5) (2007) 654–661
10. Chen, K., Gao, S., Zhu, Y., Sun, Q.: Music genres classification using text categorization method. In: MMSP. (2006) 221–224
11. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Research and Development in Information Retrieval. (1998) 275–281
12. Annesi, P., Basili, R., Gitto, R., Moschitti, A., Petitti, R.: Audio feature engineering for automatic music genre classification. In: RIAO, Pittsburgh (2007)
13. Li, T., Ogihara, M., Li, Q.: A comparative study on content-based music genre classification. In: SIGIR, NY, USA (2003) 282–289